Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3091495, 11 pages
https://doi.org/10.1155/2022/3091495

Research Article

A Detection Algorithm for Audio Adversarial Examples in EI-Enhanced Automatic Speech Recognition

Ying Huang and Jie Liu

School of Information Engineering, Xi'an University, China

Correspondence should be addressed to Ying Huang; yhuang@xawl.edu.cn

Received 15 January 2022; Accepted 19 March 2022; Published 21 April 2022

Academic Editor: Celimuge Wu

Benefiting from the development of big data, edge computing, and deep learning, splendid breakthroughs have been made in automatic speech recognition (ASR) in recent years. Since then, more and more smart products have chosen speech as the interface for human-computer interaction, which causes popularity of edge intelligence (EI) enhanced automatic speech recognition. While people are enjoying the social changes brought by speech recognition technology, a factor of instability quietly emerged called audio adversarial example which is a type of audio deliberately generated by attackers via adding subtle perturbations to the original audio signal. The added perturbations which sound like certain noise that cannot be precepted by human but will cause ASR system make wrong transcription. Three detection algorithms for audio adversarial examples are proposed in this thesis, namely, the robust detection algorithm based on WER (word error rate), the feature detection algorithm based on ADR (adversarial ratio), and the collaborative detection algorithm based on neural network. The experiment results show that three detection algorithms proposed in this thesis have a great discrimination on audio adversarial examples and achieve high AUC scores. Among them, the cooperative detection is the best and the feature detection is the worst. In addition, we found that robust detection algorithm tends to have a higher accuracy score but a lower recall score, while feature detection algorithm tends to have the converse performance. Moreover, since the proposed collaborative detection method combines the advantages of the robust detection and feature detection methods, it presents a better performance with respect to accuracy, recall, and F1 score.

1. Introduction

With the evolution of deep learning, big data, and cloud computing technologies, the accuracy of speech recognition has improved substantially. The hardware cost of speech data storage is also continually dropping. These two trends have led to more and more smart products using speech as the interface of human-computer interaction, which has resulted in more opportunities for the intelligent speech industry [1]. According to statistics from Frost and Sullivan, the market for China's intelligent speech industry has gone from only 2.87 billion yuan in 2014 to 21.65 billion yuan in 2019, at a compound annual growth rate of 53.2%. Sullivan forecasts that China's intelligent speech market will reach 65.51 billion yuan in 2023.

With the merge of edge computing and artificial intelligence, intelligent speech applications enhanced by EI are now used in scenarios such as smart homes, smart cars, smart medical devices, and smart customer service [2]. Internet companies, intelligent speech technology companies, and smart speech start-ups are all players in the global market for intelligent speech products.

The increased availability of intelligent speech devices has helped users realize the value of instinctive expression when it comes to productivity, and consequently, users' lifestyles are changing. However, speech adversarial samples have emerged as a stumbling block. The adversarial sample in speech recognition is defined as a type of audio generated by an attacker in order to deliberately add subtle disturbances to the original audio. In terms of acoustic characteristics, the speech adversarial sample has slightly more noise than the original audio, which the human ear is not sensitive to. However, it will cause an automatic speech recognition (ASR) system to transcribe errors. The widespread use of ASR systems in intelligent speech devices gives attackers more opportunities to do this. For example, attackers can generate adversarial samples against a certain type of ASR system in advance and then use social media to disseminate the adversarial samples. Adversarial samples could be input into the speech interface of a smart car, and then transcribed into a series of altered driving instructions, posing great danger to life and property. Therefore, it is very important to study the adversarial examples and defense mechanisms in speech recognition.

Nilaksh Das and Madhuri Shanbhogue et al. in [3] implemented the first interactive experimental tool, called Adagio, for audio adversarial samples, which can attack and defend the end-to-end Deep Speech model in real time visually and auditorily. In [4], Iustina Andronic et al. also discussed the possibility of MP3 compression as a defense against adversarial samples. Krishan Rajaratnam et al. [5] discussed the effect of combining audio preprocessing methods on speech classification models, using six preprocessing methods, including MP3 compression, AAC compression, bandpass filtering, and audio translation. Zhuolin Yang et al. [6] pointed out that speech is a time-domain signal with inherent time-dependent characteristics and that the introduction of antinoise can lead to the destruction of this dependence. Based on this assumption, we propose the concept of using temporal dependency (TD) for detection, which uses the ratio of the longest common prefix of partial and full transcription to the length of the entire text as a detection indicator. Tejas Jayashankar et al. [7] first proposed applying the concept of dropout [8] to the detection of audio adversarial samples. Victor Akinwande and Celia Cintas [9] introduced a novel idea, which regards the detection of adversarial samples as anomalous pattern (AP) detection in the ASR model space. The author assumes that the adversarial sample will cause abnormal activation of some nodes in the neural network. Based on this, the author uses the subset scan method to search for the most abnormal subset of data observations and then uses nonparametric scan statistics. This method quantifies the abnormality of the subset as a numerical score between 0 and 1, specifically the Berk–Jones test statistics [10] method. Qiang Zeng et al. [11] combined the fact that different ASR systems use different architectures, parameters, and training datasets to cause differences in the same audio transcription with the idea of multiversion programming [12], and proposed a novel method of adversarial sample detection, called MVP-EARS. This method uses ready-made ASR algorithms to determine whether the audio is an adversarial sample. Saeid Samizade et al. [13] proposed for the first time that the detection of adversarial samples is a binary classification problem. Based on this, this paper proposes to convolutional neural network (CNN) detection, which involves using the CNN model to train the detection method of adversarial samples and benign samples.

The main work of this paper in voice adversarial sample defense is as follows:

(1) We propose a robust detection algorithm based on word error rate (WER) which is based on the fact that adversarial samples are obtained by adding a small amount of noise to a normal sample. The algorithm detects the audio using spectral subtraction for noise reduction, then uses the WER to measure the impact of noise reduction on the audio, and then trains a classifier to differentiate adversarial samples from benign samples. The proposed approach is superior to other approach in theory and experimental results

(2) We propose a feature detection algorithm based on adversarial effect derived from the fake sample to improve the method of directly using the entire speech feature for detection. In order to characterize the adversarial nature of a certain frame and a certain speech, the proposed approach attempts to incorporate the characteristic of voice sample into neural networks, and finally, a classifier is trained to distinguish adversarial samples from the normal samples

(3) Aware of the single and linear characteristics of the above two methods, we pro-pose a neural network-based collaborative detection algorithm and introduce a binary neural network model to fit the nonlinear relationship between WER, adversarial degree, and adversarial samples, in order to further improve the security and discrimination capability of the detection algorithm. The robust detection algorithm based on WER and the calculation method based on adversarial degree are combined with the waveform characteristics of the voice itself to extract voice features. This combined method can better restore the audio itself and also provides a benign input feature for the calculation of the neural network, thereby ensuring the accuracy of the calculation results and a high recall rate

2. Related Work

2.1. Attack Model. ASR technology converts human speech into text [14–16]. From speech signals to text characters, ASR technology spans multiple basic and cutting-edge disciplines such as acoustics and linguistics, signal processing, computers, and artificial intelligence. Although research on speech recognition began as early as the 1950s, due to its complexity, the accuracy of speech recognition was not very high until the emergence of neural networks and the rise of end-to-end technology [17, 18]. Since then, the accuracy of speech recognition rate has been advancing rapidly. Compared with the traditional DNN-HMM [19] hybrid model, the end-to-end ASR system omits the steps of aligning text- and context-sensitive phonemes and can directly start training from the neural network without multiple iterations.

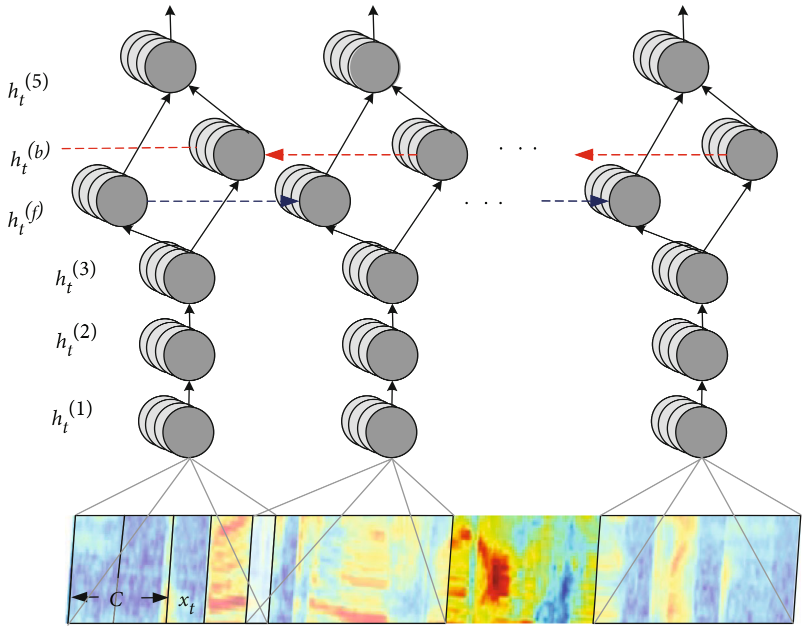

Figure 1: Deep Speech network model.

All the theoretical research and experiments in this paper are based on Deep Speech, an open source, end-to-end ASR system [20]. The network structure is shown in Figure 1. The MFCC feature of speech is used as input. The core is an RNN model with correctionist temporal classification (CTC) loss [21] as the loss function; the output is the probability distribution of the character sequence.

The Deep Speech model consists of 5 hidden layers. For input x, we use h^l to denote the l^{th} layer and h^0 to denote the input. The first 3 layers are fully connected layers. For the first layer, the input at time t is not only the characteristics x_t of time t, but also the characteristics of its front and back C frames, totaling $2C + 1$ frames. The first 3 layers are calculated by

$$h_t^l = g\left(W^l h_t^{l-1} + b^l\right), \tag{1}$$

where $g(z) = \min(\max(z, 0))$ and the maximum value is limited on the basis of ReLU, so it is also called Clipped ReLU. The fourth layer is a two-way RNN, as shown in

$$\begin{aligned} h_t^f &= g\left(W^4 h_t^3 + W_\tau^f h_{t-1}^f + b^4\right), \\ h_t^b &= g\left(W^4 h_t^3 + W_\tau^b h_{t+1}^b + b^4\right). \end{aligned} \tag{2}$$

The most common RNN is used here instead of LSTM/GRU in order to make the network structure simple and consistent and to facilitate the optimization of calculation speed. In this two-way RNN, the parameters input to the hidden unit are shared (including bias), and the RNN in each direction has its own hidden unit and hidden unit parameters. h^f is calculated from time 1 to time T, and h^b is calculated from time T in turn. The fifth layer will add the two outputs of the fourth layer bidirectional RNN as its input, as shown in

$$\begin{aligned} h_t^4 &= h_t^f + h_t^b, \\ h_t^5 &= g\left(W^5 h_t^4 + b^5\right). \end{aligned} \tag{3}$$

The last layer is a fully connected layer without activation function, which uses softmax to turn the output into a probability corresponding to each character, as shown in

$$h_{t,k}^6 = \hat{y}_{t,k} = P(c_t = k|x) = \frac{\exp\left(W_k^6 h_t^5 + b_k^6\right)}{\sum_j \exp\left(W_j^6 h_t^5 + b_j^6\right)}. \tag{4}$$

After calculating $P(c_t = k|x)$, CTC can be used to calculate $L(\hat{y}, y)$ and find the gradient of L to the parameter.

2.2. CW Attack. The CW attack is a white-box targeted attack against the ASR system, derived from the literature of Nicholas Carlini and David Wagner [22]. In this method, we propose to improve the CTC loss function y introducing the L2 norm of noise distortion and using the Adam optimizer to simultaneously optimize CTC loss and distortion to achieve a balance between distortion and CTC Loss. The loss function is shown by

$$\begin{gathered} \text{minimize } |\delta|_2^2 + c \cdot l(x + \delta, t) \\ \text{such that } \mathrm{dB}_x(\delta) \le \tau \quad , \\ \mathrm{dB}_x(\delta) = \mathrm{dB}(\delta) - \mathrm{dB}(x) \end{gathered} \tag{5}$$

where δ is the added noise; c is the weight; x is the original audio; l is the CTC loss function; t is the target transcription text; $\mathrm{dB}_x(\delta)$ is the distortion of the noise δ relative to the original waveform x, measured in decibels (dB); τ is the maximum distortion constant; and $\mathrm{dB}(\cdot)$ is the logarithmic scale which is used to measure the relative loudness of audio or noise samples; the calculation method is shown in

$$\mathrm{dB}(x) = \max_i 20 \cdot \log_{10}(x_i), \tag{6}$$

where x_i represents the value of the i^{th} sampling point of the waveform x.

Why can CW attacks be used to generate adversarial examples? It should be noted that the CTC loss function reflects the relationship between the target transcription text and its corresponding audio. When the audio does not match the text, the CTC loss is larger. Therefore, reducing the CTC loss of the audio to is equivalent to increasing the CTC loss of the audio and the original transcribed text, but the direction of its increase is to approach. In addition, to be able to converge as quickly as possible, we use the fast gradient thinking in the FGSM algorithm. In each iteration, loss is used to derive the added noise to obtain the gradient that makes the loss function change the fastest, and the disturbance noise is "updated" along this gradient direction until the transcription target is reached.

Figure 2 shows a comparison of the audio waveform before and after the CW attack. The original audio content is "but everything had changed," and the CW attack is successfully transcribed as "nothing is impossible." It is clear that the CW attack modifies the entire waveform. The pronunciation segment of the adversarial sample has a greater similarity to the original waveform, while the silent segment has a significant gap. In addition, compared with the original audio, we can hear obvious TV-like snowflake noise.

3. Algorithms

In this section, we introduce our proposed audio adversarial sample detection model. As shown in Figure 3, the model includes seven components.

(i) Speech interface module: Corresponding to the upper left corner of Figure 3, this module is responsible for detecting the legality of input audio files, that is, whether the sampling rate and format meet the specifications, and then converting the speech into the form of a one-dimensional vector

(ii) Noise reduction module: The core of the robust detection algorithm, this module is responsible for noise reduction and storage of audio. Here, it imitates the code style of the audio adversarial sample attack library and encapsulates all the noise reduction-related code into the denoise.py file. Called through the interface of the denoise input audio, the function returns the save path of the audio after noise reduction

(iii) Feature extraction module: This module, which forms the core of the feature detection algorithm, is responsible for extracting the feature vector of the filter banks of the audio. In addition, the module is responsible for the extraction of MFCC features; that is, another DCT operation is performed on the basis of the filter banks for Deep Speech voice recognition system input

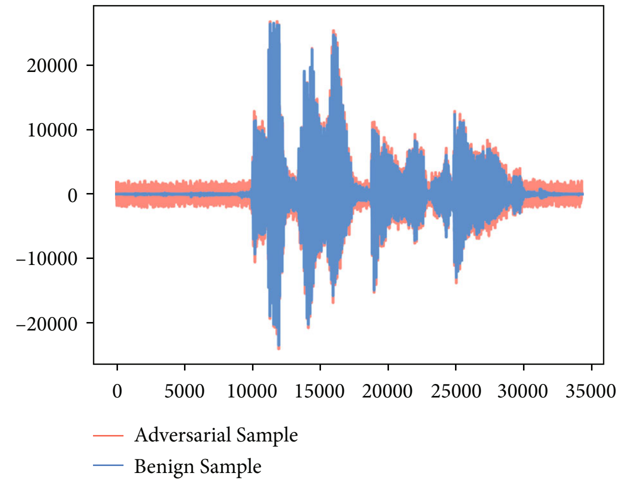

Figure 2: Audio waveform comparison before and after CW attack.

(iv) Speech recognition system: This system is responsible for transcribing the input MFCC features into human-understandable text

(v) WER calculation module: (described in Section 3.1)

(vi) Adversity calculation module: (described in Section 3.2)

(vii) Two-class neural network: (described in Section 3.3)

3.1. Robust Detection Algorithm Based on Word Error Rate.

The core of the robust detection algorithm based on WER is spectral subtraction noise reduction. In this method, the first spectral subtraction noise reduction is performed on the audio to be detected [23], and then, the audio is transcribed before and after noise reduction through the ASR system to calculate the WER of the audio to be detected. Finally, according to the differentiation of adversarial samples based on WER, a classifier is designed for detection. The algorithm principle and process are as follows:

According to the generation process of adversarial samples, let $y(n)$ be an audio adversarial sample with added antinoise, and then $y(n)$ is composed of original audio $x(n)$ and additive noise $d(n)$; that is, the form of the additive model is shown in

$$y(n) = x(n) + d(n). \tag{7}$$

The Fourier transform on both sides of the equation is shown in

$$Y(\omega) = X(\omega) + D(\omega). \tag{8}$$

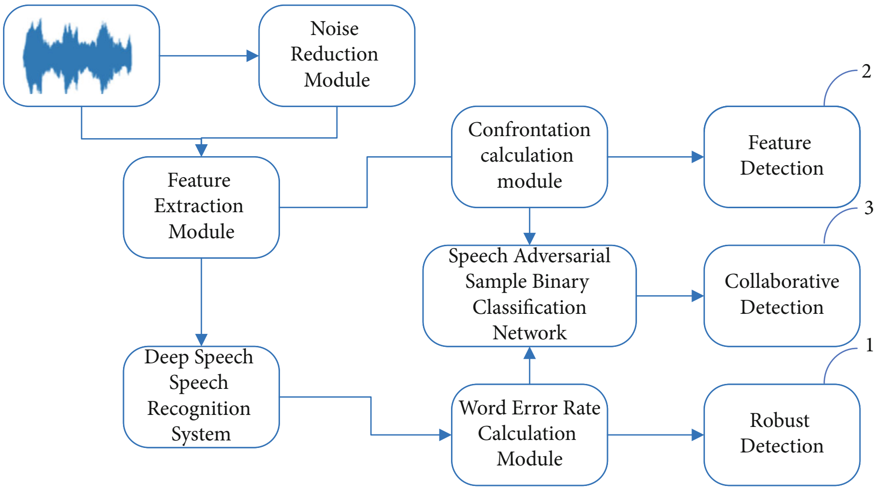
Noise Reduction Module
Feature Extraction Module
Deep Speech Speech Recognition System
Confrontation calculation module
Speech Adversarial Sample Binary Classification Network
Word Error Rate Calculation Module
Feature Detection
Collaborative Detection
Robust Detection
2
3
1

Figure 3: Speech adversarial sample detection model.

If expressed by the power spectrum, the form of the additive model is shown in

$$|Y(\omega)|^2 = |X(\omega)|^2 + |D(\omega)|^2 + 2\,\mathrm{Re}\left\{X(\omega)\overline{D(\omega)}\right\}. \quad (9)$$

Here, $2\,\mathrm{Re}\,\{X(\omega)\overline{D(\omega)}\}$ is called the cross term. Due to the vibration of the vocal organs, the speech signal is usually nonstationary. But if only one of the frames is intercepted, assuming that it is 10 to 30 ms, the speech in this frame has a stationary characteristic. In the same way, the noise signal is also stable or slowly changing at the microscopic scale. Therefore, it is considered that the mean value of the additive noise $d(n)$ is 0, and is not related to $x(n)$; that is, the cross term is 0. The above formula is simplified as

$$|X(\omega)|^2 = |Y(\omega)|^2 - |D(\omega)|^2. \quad (10)$$

In the speech signal, it is generally considered that there is no speech activity in the first few frames, so the first few frames can be regarded as pure noise signals; that is, the noise spectrum $|D(\omega)|^2$ can be estimated using these frames. Because the phase of the speech signal will not affect humans' understanding of speech, after obtaining the amplitude spectrum of the original audio, the phase of the speech adversarial sample can be used to approximate the speech phase of the original audio. At this time, an approximation original audio can be obtained in theory.

Further, we use the audio before and after noise reduction to obtain the reference text and the predicted text through the ASR system. We hope that the impact of noise reduction can be reflected in the differences in the text. To measure the inconsistency between two paragraphs of text, this method uses WER.

WER is generally used to compare predicted text and reference text in units of characters and to quantify the difference between the two texts. It is typically used to measure the performance of an ASR system and is a key indicator in the field of speech recognition. Its calculation formula is shown in

$$\mathrm{WER} = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}. \quad (11)$$

Note that S is the number of words that need to be replaced in the reference text, D is the number of words that need to be deleted in the reference text, I is the number of words that need to be inserted into the reference text, and C is the correct number of words in the reference text. As a result, $N = S + D + C$ is the character length of the reference text.

The numerator of the WER is equivalent to the edit distance of two paragraphs of text [24]. Editing distance is defined as the minimum operation required to change from one text to another. The executable operations include replacing a character, deleting a character, and inserting a character. The industry has a classic dynamic programming solution to this problem, and its state transition equation is shown in

$$\mathrm{DP}_{i,j} = \min\begin{pmatrix} \mathrm{DP}_{i-1,j-1} + 0 & \text{if } h_i = r_j \\ \mathrm{DP}_{i-1,j-1} + 1 & (\text{Substitution}) \\ \mathrm{DP}_{i,j-1} + 1 & (\text{Insertion}) \\ \mathrm{DP}_{i-1,j} + 1 & (\text{Deletion}) \end{pmatrix}, , \quad (12)$$

where h is the predicted text, r is the reference text, and DP is the matrix used for state transition which dimension is $|h| \times |r|$.

3.2. Feature Detection Algorithm Based on Adversarial Degree. The core of the feature detection algorithm based on adversarial degree is the extraction and application of filter banks features. In this method, we first extract the filter banks of the audio to be detected and then calculate the antagonism of the audio to be detected based on the filter banks. Finally, according to the degree of discrimination of adversarial samples, a classifier is designed for detection. The algorithm principle and process are as follows.

3.2.1. Pre-Emphasis. The first step of the algorithm is to apply a pre-emphasis filter to the signal. Compared with the low frequency, the high frequency usually has a smaller amplitude, so the pre-emphasis filter can be used to balance the spectrum and amplify the high frequency. In addition, the pre-emphasis can also avoid numerical problems during the Fourier transform operation and improve the signal-to-noise ratio (SNR). The specific calculation formula is shown in

$$y(t) = x(t) - \alpha x(t-1), \tag{13}$$

where x is the speech signal, y is the signal after pre-emphasis, and α is the pre-emphasis coefficient, which is generally selected as 0.95 or 0.97.

3.2.2. Framing. After pre-emphasis, the signal needs to be divided into short-time frames. Under normal circumstances, the frequency in the speech signal is not static, and the Fourier transform of the entire speech signal will lose the frequency contour of the signal. Therefore, the signal is also processed in units of frames, and then, the approximate value of the signal frequency profile is obtained by merging adjacent frames. In speech processing, the frame size is usually set to 25 ms.

3.2.3. Window Adding. After the signal is cut into frames, a window function such as a Hamming window needs to be applied to each frame to offset the assumption of unlimited data made by the fast Fourier transform (FFT) and reduce spectrum leakage. The form of the Hamming window is shown in

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \tag{14}$$

where $0 \le n \le N - 1$ and N is the length of the window.

3.2.4. Fourier Transform and Power Spectrum. Next, we perform short-time Fourier transform on each frame, which is also called N-point FFT. N is usually 256 or 512. The power spectrum calculation formula is shown in

$$P = \frac{|\mathrm{FFT}(x_i)|^2}{N}, \tag{15}$$

where x_i is the i^{th} frame audio signal.

3.2.5. Filter Banks. Finally, the Mel-level triangular filter (usually 40 filters) is applied to the power spectrum to extract the filter banks features. The Mel scale imitates the human ear's perception of sound; that is, it has a higher discriminative power at a lower frequency and a lower discriminatory power at a higher frequency. The conversion formula of Hertz f and Mel m is shown in

$$\begin{aligned} m &= 2595 \log_{10}\left(1 + \frac{f}{700}\right), \\ f &= 700\left(10^{m/2595} - 1\right). \end{aligned} \tag{16}$$

Each filter in the Mel filter bank is triangular, with a response of 1 at the center frequency, and linearly decreases to 0 toward the center frequency of the adjacent filters. The filters in the Mel filter bank are shown in

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k < f(m) \\ 1 & k = f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) < k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}, \tag{17}$$

where m is the subscript of the filter bank; in this method $1 \le m \le 40$, $f(m)$ is the center frequency of the m^{th} triangular filter, and $H_m(k)$ represents the response of the m^{th} triangular filter at k Hz. Because the human ear's perception of sound is not linear, it is necessary to use the log function for nonlinear processing at the end.

3.2.6. Confrontation. Statistical observation of the filter banks of benign and adversarial samples reveals that the probability of positive values in the filter banks features of adversarial samples is significantly higher than that of benign audio samples. In addition, the longer the noise duration, the greater the noise amplitude, and the greater the probability of a positive value. Based on this, we propose the concept of adversarial frames. Frames with nonnegative filter banks feature values are regarded as adversarial frames, which indicate the degree of disbelief in this frame.

Furthermore, the speech signal is counted in units of frames in filter banks, and the concept of adversarial degree is proposed, which represents the proportion of adversarial frames in the audio. The greater the proportion, the greater the degree of disbelief in the audio, that is, the greater the possibility of treating it as a confrontational sample. The smaller the proportion, the more authentic the audio is. The calculation method of antagonism is shown in

$$\mathrm{ADR} = \frac{\sum_i (f \ge 0 \forall f \in \mathrm{fea}_i)}{N}, \tag{18}$$

where fea is the feature matrix of the audio filter banks and N is the first dimension of fea, which is related to the audio duration.

3.3. Collaborative Detection Algorithm Based on Neural Network. All single detection algorithms may bring about the problem of insufficient robustness, and the algorithm proposed in this paper is no exception. An attacker can deliberately reduce a single index to carry out more advanced secondary attacks. In addition, the binary classification in the real scene is usually not linearly separable, and all the methods that use linear classification will inevitably result in the lack of a certain performance index of accuracy or recall. Therefore, to further improve the robustness of the model and the algorithm's discrimination against adversarial samples, we combine the methods proposed in Sections 3.1 and 3.2 to provide a better detection method.

Here, we regard WER and adversarial degree as the characteristics of artificially extracted speech samples and then use neural network for nonlinear fitting training to achieve the effects of classification and detection.

In this paper, a lightweight binary neural network is selected. In addition to the input layer and the output layer, it only includes two hidden layers and the corresponding dropout layer, as shown in Table 1.

Table 1: Two-class neural network architecture.

Layer (type)	Output shape
dense_1 (dense)	(None, 64)
activation_1 (activation)	(None, 64)
dropout_1 (dropout)	(None, 64)
dense_2 (dense)	(None, 64)
activation_2 (activation)	(None, 64)
dropout_2 dDropout)	(None, 64)
dense_3 (dense)	(None, 1)
activation_3 (activation)	(None, 1)

4. Implementation and Evaluation

4.1. Databases. The Common Voice [25] corpus is an initiative from Mozilla, which contains six files with tab-separated values (TSV files) and a single clips subdirectory that contains all of the audio data, where each of the six TSV files represents a different segment of the voice data, with all six having the following column headers: [client_id, path, sentence, up votes, down_votes, age, gender, accent]. It is a collection of self-recorded voices uploaded by many users on the Common Voice website. The text content comes from many public domains, such as blog posts submitted by users, old books, movies, and other public speeches. According to Mozilla, the main purpose of the project is to train and test the ASR system. The goal is to help teach machines how to speak, but Mozilla also encourages its use for other purposes.

The Common Voice corpus is divided into three parts. The "valid" subset is the audio that has been heard by at least two people and that most of the listeners think matches the text. The "invalid" subset contains the audios that do not match their corresponding text judged by at least 2 persons. And the remaining audios form the subset named "other". Furthermore, "valid" and "other" are divided into three parts: "dev" is used for development and experimentation, "train" is used for speech recognition training, and "test" is used for testing WER.

Considering the cost in time and hardware, this paper finally selected 8071 audio files of "cv-valid-dev" and "cv-valid-test" as the preliminary screening of the dataset.

The audio files of the Common Voice corpus are all in .mp3 format. Therefore, first, the format conversion of the preliminary audio files is required, and then Deep Speech is used for transcription, and the WER index is tested. According to the results, Deep Speech reached an average WER of 7.33% and performed well on the preliminary screening dataset. Finally, this paper screened out 1,200 speech samples with a WER of 0 and the length of the transcribed text not exceeding 57 as the experimental benign sample dataset and used the CW attack to generate the corresponding adversarial sample dataset. If the length of the transcribed text of the benign sample does not exceed 37, the CW attack target is set to "nothing is impossible"; otherwise, the attack target is set to "if winter comes can spring be far behind?"

4.2. Environment. The hardware environment of the experiment in this article is shown in Table 2.

The software environment of the experiment in this paper is shown in Table 3.

In Table 3, Deep Speech code is Mozilla's code implementation of Deep Speech's speech recognition model, and Deep Speech model is a trained model file that stores the weights, biases, gradients, and other variable values of the model. We need to pay attention to compatibility when using the Python package. The adapted version number is given here. CUDA and cuDNN are drivers that need to be installed when using Nvidia graphics cards. We can also install the TensorFlow version, if it is the correct version.

4.3. Indicators. In order to better introduce the two-category index, a confusion matrix is first introduced here. The confusion matrix is shown in Table 4.

4.3.1. ACC. ACC indicates the proportion of samples with correct predictions to the total samples. The ACC calculation formula is shown in

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{19}$$

4.3.2. AUC. The area under the curve (AUC) represents the area under the ROC curve. Here, in order to better understand the ROC curve, we first introduce the true positive rate (TPR) and false positive rate (FPR). TPR represents the proportion of all positive samples in the dataset that are correctly predicted. The calculation formula is shown in

$$\mathrm{FPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{20}$$

Table 2: Hardware environment.

Items	Parameter
CPU	Intel Xeon E3-1230v5
RAM	16GB DDR4
GPU	NVIDIA Quadro K420
Storage	1 T SSD

Table 3: Software environment.

(a)

Component	Version
Ubuntu	16.04
Python	3.5.2
Deep Speech code	0.4.1
Deep Speech model	0.4.1
CUDA	9.0
cuDNN	7.0

(b)

Python packages	Version
pandas	0.24.0
numpy	1.16.4
Keras	2.2.4
ds-ctcdecoder	0.4.1
tensoflow-gpu	1.12.0
scipy	1.4.1

Table 4: Confusion matrix.

		Actual result	
		P	N
Prediction	P	TP	FP
	N	FN	TN

where T and F represent true and false, and P and N represent positive and negative.

FPR represents the proportion of negative samples that are predicted to be positive samples. The calculation formula is shown in

$$\mathrm{FPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{21}$$

Every time a threshold is set, a set of TPR and FPR values can be obtained. Therefore, the score of each sample in the test set is set as a threshold, so that multiple sets of TPR and FPR values can be obtained. At this time, FPR is used as the abscissa and TPR as the ordinate to draw the ROC curve.

4.3.3. Precision. Precision represents the proportion of positive samples that are correctly predicted to all positive samples. The calculation formula is shown in

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{22}$$

4.3.4. Recall. Recall represents the proportion of positive samples that are correctly predicted to all positive samples. It basically has the same meaning as the true rate, except that the name is different. The calculation formula is shown in

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{23}$$

4.3.5. F1 Score. F1 score is defined as the harmonic mean of precision and recall. The calculation formula is shown in

$$F_1 = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{24}$$

4.4. Result and Analysis. Figures 4 and 5 show the comparison diagrams of WER and adversarial degree distribution of benign samples and adversarial samples generated by CW attacks. According to the WER distribution map, it is clear that the WER of the benign samples is concentrated in the range of 0 to 0.1, while the WER of nearly every adversarial sample is greater than 0.1. Furthermore, according to the adversarial degree distribution map, it is clear that the WER of the adversarial samples is concentrated in the range of 0.9 to 1.0, while the benign samples have a wider distribution range, but most of them are less than 0.9. Therefore, the WER and adversarial degree have a good degree of success in differentiating the adversarial samples generated by the CW attack. In terms of the distribution ratio, the differentiation capability of adversarial degree is slightly weaker than that of the WER.

Figure 6 shows the joint distribution diagram of WER and adversarial degree of the benign sample and the adversarial sample generated by the CW attack. The boundary between the benign sample and the adversarial sample is relatively clear, with points crossing only sporadically. It can be concluded that the collaborative detection algorithm is very successful at differentiating the adversarial samples generated by the CW attack.

Figure 7 shows the ROC curves of the three detection algorithms against CW attacks. The upper left corners of the three curves are infinitely close to the $(0, 1)$ point, and the AUC value is greater than 0.99, indicating that these three algorithms perform very well in detecting CW attacks. In addition, the ROC curve of the collaborative detection completely covers the other two curves, indicating that the performance of the collaborative detection algorithm is better than that of a single detection. Because a single detection has a high degree of discrimination against the adversarial samples generated by the CW attack, the improvement of the discrimination degree by coordinated detection is limited. The ROC curves of robust detection and feature

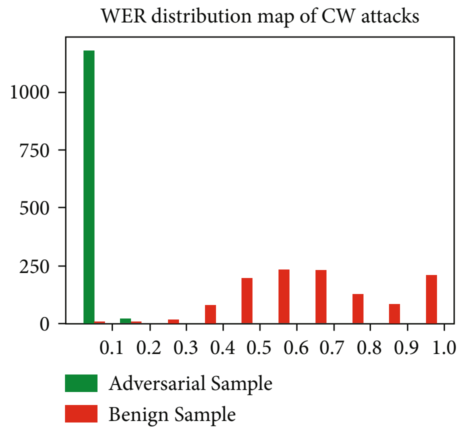

Figure 4: WER distribution map of CW attacks.

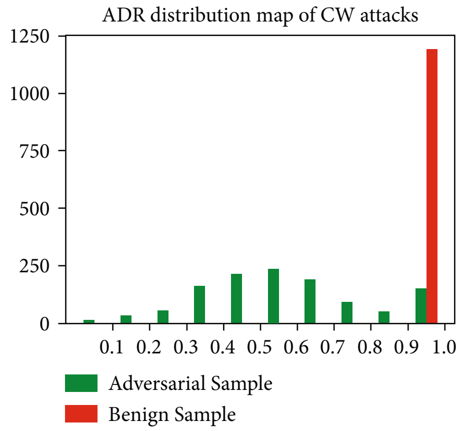

Figure 5: ADR distribution map of CW attacks.

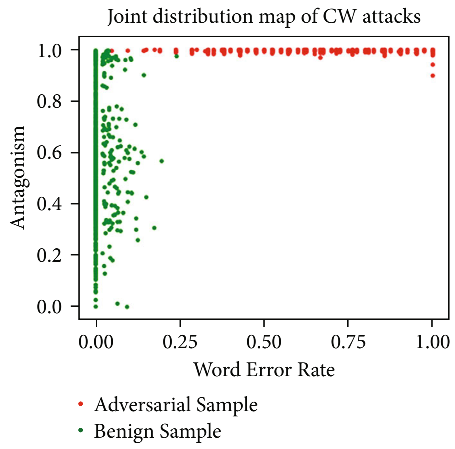

Figure 6: Joint distribution map of CW attacks.

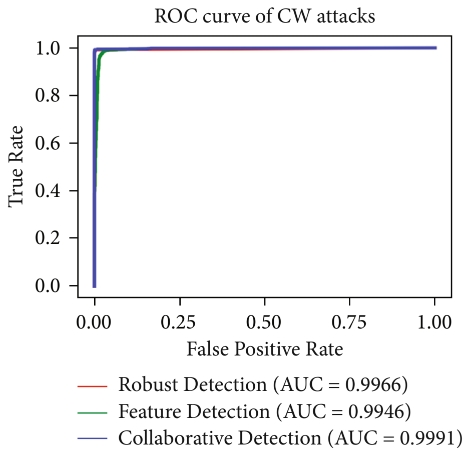

Figure 7: ROC curve of CW attacks.

Table 5: Indicators of CW attacks.

	Robust detection	Feature detection	Collaborative detection
ACC	0.9950	0.9817	*0.9967*
AUC	0.9966	0.9946	*0.9991*
Precision	0.9983	0.9738	*0.9983*
Recall	0.9667	0.9900	*0.9967*
F1 score	0.9822	0.9818	*0.9975*

detection overlap. In terms of the AUC value, the robust detection performs the best.

Next, we will further discuss the algorithm in terms of its specific performance on the test set, that is, ACC, accuracy, recall, and F1 score. Table 5 shows the CW attack detection indicators. By comparing the three detection algorithms, we can draw three conclusions.

(1) The three detection algorithms all have a good degree of discrimination against adversarial samples in CW attack scenarios, with the collaborative detection algorithm the best, followed by the robust detection and the feature detection. Because the collaborative algorithm also detects the resistance robust detection characteristics, such as WER, and the feature-sensitive characteristics of neural networks, its success rate is higher than that of the other two. Feature changes often affect the robustness of the system, so in terms of index values, robustness, and detection methods are better than feature detection methods

(2) Robust detection algorithms based on WER tend to have a higher accuracy rate, but the recall rate is low. Feature detection algorithms based on adversarial degree tend to have a higher recall rate, but the accuracy rate is low. It shows that the robust detection algorithm based on the suberror rate has higher accuracy in the retrieval accuracy rate than the feature detection based on adversarial degree. This also confirms the conclusion (1) from another aspect, that is, the robust detection method better than feature detection methods

(3) The collaborative detection algorithm based on neural network improves the discrimination capability

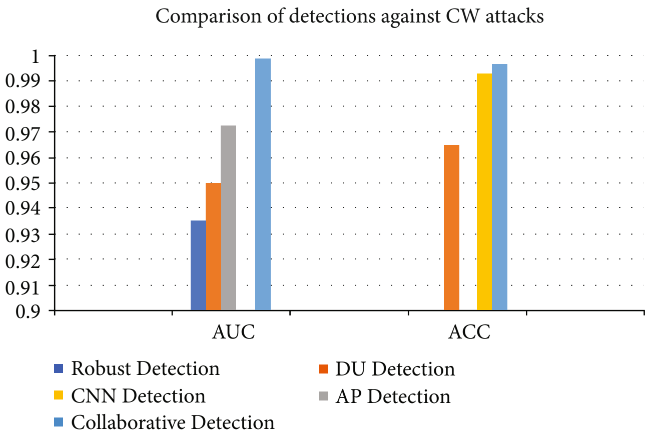

Figure 8: Comparison of detections against CW attacks.

of voice adversarial samples more than single detection. In addition, the collaborative detection algorithm integrates the advantages of robust detection and feature detection, making it have a higher accuracy rate and recall rate, as well as a more balanced detection capability

4.5. Comparisons. In this section, we compare the best performing collaborative detection algorithm with the existing detection algorithms, which are TD detection, DU detection, AP detection, and CNN detection. The comparison results are shown in Figure 8. Where there is no bar, it indicates that the author did not use the corresponding indicators.

Figure 8 shows that the detection scheme proposed in this paper achieves higher scores in both AUC and ACC indicators than the existing detection schemes, indicating that the collaborative detection algorithm has a stronger capability to detect CW attacks.

5. Conclusions

In recent years, thanks to China's favorable policies for artificial intelligence and the relatively mature technologies of speech recognition, big data, and cloud computing, the country's intelligent voice industry has experienced a period of rapid development. However, the popular ASR systems are suffering from the severe threat of audio adversarial samples. Adding even a slight disturbance to original audios, that is difficult to be detected by listeners, will make these systems output erroneous transcriptions. This poses a serious threat to the security of smart voice devices, which is the focus of this article's research.

This paper proposes three detection schemes for detecting adversarial samples: a robust detection algorithm based on word error rate, a feature detection algorithm based on adversarial degree, and a collaborative detection algorithm based on neural network. The robust detection algorithm is based on WER from the perspective of generating voice confrontation samples. It proposes the idea of using spectral subtraction noise reduction to destroy the artificially added perturbation in the confrontation sample and then uses WER as a measurement standard for detection. From the perspective of voice features, the feature detection algorithm based on adversarial degree proposes two concepts: detecting the filter banks feature of the speech frame as a unit and adversarial frame and adversarial degree. The algorithm uses these as the detection criteria. Considering the problems that might be caused by single linear detection, the neural network-based collaborative detection algorithm combines WER and adversarial degree to jointly detect voice adversarial samples by training a neural network.

The experimental results show that all three detection algorithms display good discrimination against CW attacks, with the collaborative detection performance the best, followed by robust detection and then feature detection. The results also show that robust detection algorithms tend to have higher accuracy, but the recall rate is low. The feature detection algorithms tend to have higher recall, but the accuracy is low. The collaborative detection algorithm integrates the advantages of robust detection and feature detection. While improving the overall discrimination, it also has a higher accuracy rate and recall rate, as well as a more balanced detection ability, which proves the necessity of joint detection.

Although the results show that the research in this paper has achieved good results, it should be noted that the adversarial samples studied in this paper are directly input to the voice interface and cannot form an attack effect after being broadcast in the air. However, the latest work [26, 27] shows that although there are many restrictions, there are already adversarial samples that can be played and then attacked. Therefore, it is important to continue research on the defense of voice adversarial samples. In addition, the author believes that it is very valuable to use each frame of speech as a unit of detection, but due to time limitations, this could not be addressed in this paper. Future research will explore this.

Data Availability

The Common Voice [25] corpus is an initiative from Mozilla. It is a collection of self-recorded voices uploaded by many users on the Common Voice website. The text content comes from many public domains, such as blog posts submitted by users, old books, movies, and other public speeches. According to Mozilla, the main purpose of the project is to train and test the ASR system. The goal is to help teach machines how to speak, but Mozilla also encourages its use for other purposes.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[2] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE*

Transactions on Intelligent Transportation Systems, pp. 1–14, 2022.

[3] N. Das, M. Shanbhogue, S. T. Chen, L. Chen, M. E. Kounavis, and D. H. Chau, "Adagio: Interactive experimentation with adversarial attack and defense for audio," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 677–681, Springer, Cham, 2018.

[4] I. Andronic, L. Kürzinger, E. R. Chavez Rosas, G. Rigoll, and B. U. Seeber, "MP3 compression to diminish adversarial noise in end-to-end speech recognition," in *Speech and Computer. SPECOM 2020. Lecture Notes in Computer Science, vol 12335*pp. 22–34, Springer, Cham.

[5] K. Rajaratnam, K. Shah, and J. Kalita, "Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition," http://arxiv.org/abs/1809.04397.

[6] Z. Yang, B. Li, P. Y. Chen, and D. Song, "Characterizing audio adversarial examples using temporal dependency," http://arxiv.org/abs/1809.10875.

[7] T. Jayashankar, J. L. Roux, and P. Moulin, "Detecting audio attacks on ASR systems with dropout uncertainty," http://arxiv.org/abs/2006.01906.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] V. Akinwande, C. Cintas, S. Speakman, and S. Sridharan, "Identifying audio adversarial examples via anomalous pattern detection," http://arxiv.org/abs/2002.05463.

[10] R. H. Berk and D. H. Jones, "Goodness-of-fit test statistics that dominate the Kolmogorov statistics," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 47, no. 1, pp. 47–59, 1979.

[11] Q. Zeng, J. Su, C. Fu et al., "A multiversion programming inspired approach to detecting audio adversarial examples," in *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pp. 39–51, IEEE, Portland, OR, USA, 2019.

[12] A. Avizienis and L. Chen, "On the implementation of N-version programming for software fault-tolerance during program execution," in *International Computer Software and Applications Conference (COMPSAC)*, pp. 149–155, IEEE, Chicago, USA, 1977.

[13] S. Samizade, Z. H. Tan, C. Shen, and X. Guan, "Adversarial example detection by classification for deep speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3102–3106, Barcelona, Spain, 2020.

[14] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th International Web for All Conference*, pp. 1–8, New York, 2016.

[15] H. Ibrahim and A. Varol, "A study on automatic speech recognition systems," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–5, Beirut, Lebanon, 2020.

[16] X. Lu, S. Li, and M. Fujimoto, "Automatic speech recognition speech-to-speech translation," in *SpringerBriefs in Computer Science*, Y. Kidawara, E. Sumita, and H. Kawai, Eds., pp. 21–38, Springer, Singapore, 2020.

[17] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, p. 1018, 2019.

[18] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5060–5064, Shanghai, China, 2016.

[19] J. Novoa, J. Wuth, J. P. Escudero, J. R. Fredes, R. Mahu, and N. B. Yoma, "DNN-HMM based automatic speech recognition for HRI scenarios," in *The 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 150–159, IEEE/ACM, Chicago, USA, 2018.

[20] A. Hannun, C. Case, J. Casper et al., "Deep speech: scaling up end-to-end speech recognition," http://arxiv.org/abs/1412.5567.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, New York, 2006.

[22] N. Carlini and D. Wagner, "Audio adversarial examples: targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, San Francisco, CA, USA, 2018.

[23] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 208–211, Washington, DC, USA, 1979.

[24] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[25] R. Ardila, M. Branson, K. Davis et al., "Common voice: a massively-multilingual speech corpus," http://arxiv.org/abs/1912.06670.

[26] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning (ICML)*, pp. 5231–5240, PMLR, Long Beach, CA, USA, 2019.

[27] L. Schönherr, T. Eisenhofer, S. Zeiler, T. Holz, and D. Kolossa, "Imperio: robust over-the-air adversarial examples for automatic speech recognition systems," in *Annual Computer Security Applications Conference*, pp. 843–855, New York, 2020.